

# PROBABILISTIC SIGNATURE SCHEME

This application is based on Provisional Application Serial No. 60/037,530, filed February 10, 1997.

## BACKGROUND OF THE INVENTION

### 5 Technical Field

The present invention relates generally to digital signature schemes and, more particularly, to an RSA-based signing scheme that combines excellent efficiency with attractive security properties.

### 10 Brief Description of the Related Art

In the RSA public key system, a party has public key  $(N, e)$  and secret key  $(N, d)$ , where  $N$  is a  $k$ -bit modulus, the product of two  $(k/2)$ -bit primes, and  $e, d \in \mathbb{Z}_{\phi(N)}^*$  satisfy  $ed \equiv 1 \pmod{\phi(N)}$ . The RSA function  $f: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$  is defined by  $f(x) = x^e \pmod{N}$  and its inverse  $f^{-1}: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$  is defined by  $f^{-1}(y) = y^d \pmod{N}$  ( $x, y \in \mathbb{Z}_N^*$ , where  $\mathbb{Z}_N^*$  denotes the set of numbers between 1 and  $N-1$  which are relatively prime to  $N$ ). The function  $f$  can be used for encryption and  $f^{-1}$  for decryption. The generally-made assumption is that  $f$  is trapdoor one-way; roughly, if one does not know  $d$  (or the prime factors of  $N$ ), then it is hard to compute  $x = f^{-1}(y)$  for a  $y$  drawn randomly from  $\mathbb{Z}_N^*$ .

A widely employed paradigm to sign a document  $M$  is to first compute some "hash"  $y = \text{Hash}(M)$  and then set the signature to  $x =$

$f^{-1}(y) = y^d \bmod N$ . To verify that  $x$  is a signature of  $M$ , one computes  $f(x) = x^e \bmod N$  and checks that this equals  $\text{Hash}(M)$ . This technique is the basis for several existing standards. A necessary requirement on  $\text{Hash}$  in such a scheme is that it be collision-intractable and produce a  $k$ -bit output that encodes a point in  $\mathbb{Z}_N^*$ . Accordingly,  $\text{Hash}$  is most often implemented via a cryptographic hash function like  $h = \text{MD5}$  (which yields a **128** bit output and is assumed to be collision-intractable) and some padding. A concrete example of such a scheme is described in PKCS #1: RSA Encryption Standard (Version 1.4), June 1991, and PKCS #7, Cryptographic Message Syntax Standard (Version 1.4), June 1991, RSA Data Security, Inc., where the hash is:

$$\text{Hash}_{\text{PKCS}}(M) = \text{0x0001FFFF} \dots \text{FFFF00} \mid h(M).$$

In the above expression, the "0x" indicates that the following number is written in hexadecimal notation, and "|" denotes concatenation. Such a signature scheme may be called a "hash-then-decrypt" scheme.

The security of a hash-then-decrypt signature depends on how  $\text{Hash}$  is implemented. But the security of a scheme like  $\text{Sign}_{\text{PKCS}}(M) = f^{-1}(\text{Hash}_{\text{PKCS}}(M))$  cannot be justified given only that RSA is trapdoor one-way, even under the assumption that hash function  $h$  is ideal. This is because the set of points  $\{\text{Hash}_{\text{PKCS}}(M) : M \in \{0,1\}^*\}$  has size at most  $2^{128}$  and hence is a very sparse, and a very structured, subset of  $\mathbb{Z}_N^*$ . This lack of

demonstrable security is disadvantageous. In particular, although there is no known attack on this scheme, it is preferable to have a signature scheme with some proof of security. The same issue arises for other known standards, including ISO/IEC 9796. There, the hash function involves no cryptographic hashing, and the message  $M$  is easily recovered from  $Hash(M)$ .

Thus, the security of the current PKCS standards, as well as that of the ISO standard, cannot be justified based on the assumption that RSA is trapdoor one-way. Other standards, such as described in *Privacy Enhancement for Internet Electronic Mail: Part III Algorithms, Modes, and Identifiers*, by Balenson, IETF RFC 1423, February, 1993, are similar to the RSA standard, and the same reasoning applies.

Signature schemes whose security can be provably based on the RSA assumption include the schemes described in the following representative publications: Goldwasser, Micali and Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM Journal of Computing, 17(2):281-308, April 1988; Bellare and Micali, *How to sign given any trapdoor permutation*, JACM Vol. 9, No. 1, 214-233, January, 1992; Naor and Yung, *Universal one-way hash functions and their cryptographic applications*, Proceedings of the 21st Annual Symposium on Theory of Computing, ACM, 1989; Rompel, *One-way Functions are Necessary*

and Sufficient for Secure Signatures, Proceedings of the 22nd Annual Symposium on Theory of Computing, ACM, 1990; and Dwork and Noar, *An efficient existentially unforgeable signature scheme and its applications*, Advances in Cryptology - Crypto 94 Proceedings, Lecture Notes in Computer Science Vol. 839, Y. Desmedt. ed., Springer-Verlag, 1994. The major advantage of these works is that they can be proven to be sound, under some formalized mathematical assumption. On the other hand, these are not practical schemes; their cost (in computation time and storage) is so high that they are not considered for real world security applications.

There are additional signature schemes that have been proven secure under the assumption that a hash function which they use behaves as though it were a random function. Such schemes can be based on the hardness of factoring, or on other assumptions. Some of these schemes have been derived from identification schemes, as was first described by Fiat and Shamir, *How to prove yourself: practical solutions to identification and signature problems*, Advances in Cryptology - Crypto 86 Proceedings, Lecture Notes in Computer Science, Vol. 263, A. Odlyzko ed., Springer-Verlag, 1986. The efficiency of those schemes varies. The computational requirements are often lower than a hash-then-decrypt RSA signature, although key sizes are typically larger.

The paradigm of protocol design using hash functions that are regarded (in proofs) as random functions is thus well-developed, as described in Bellare and Rogaway, *Random oracles are practical: a paradigm for designing efficient protocols*, Proceedings of the First Annual Conference on Computer and Communications Security, ACM, **1993**; and Bellare and Rogaway, *Optimal Asymmetric Encryption*, Advances in Cryptology - Eurocrypt **94** Proceedings, Lecture Notes in Computer Science Vol. **950**, A. De Santis ed., Springer-Verlag, **1994**.

When a signature scheme is proven secure, the security proof demonstrates how to transform an attack on the signature scheme into an attack on the underlying mathematical primitive. For example, in a scheme based on factoring numbers, the proof would show how to turn a forging algorithm against the signature scheme into a factoring algorithm. The efficiency of this "reduction" quantifies the demonstrated security. A signature scheme is said to have "tight" demonstrated security if it has been proven secure by a highly efficient reduction. Tight demonstrated security is desirable because for such a scheme the security parameter (e.g., length of RSA modulus) which is deemed adequate for the mathematical primitive is necessarily adequate for the signature scheme, too.

None of the prior art has taught a signature scheme with tight demonstrated security based on a simple construction.

There remains a need in the art to provide new signature schemes that fall in the hash-then-decrypt paradigm, so that they are easy to implement, yet are simple, efficient, and practical, and, above all, have attractive security properties like tight  
5 demonstrated security. The present invention addresses this need.

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228

## BRIEF SUMMARY OF THE INVENTION

An object of the present invention is to provide new signature schemes that are as simple and efficient as standardized ones. Assuming the underlying hash function is ideal, the inventive methods are not only provably secure, but provably secure in a strong sense. In one embodiment involving RSA, signing takes one RSA decryption plus some hashing, verification takes one RSA encryption plus some hashing, and the size of the signature is the size of the modulus. The security of the inventive scheme in this embodiment is tightly related to the security of the RSA function.

The inventive teachings are also extended to provide schemes for Rabin signatures with analogous properties; in particular, their security can be tightly related to the hardness of factoring.

It is known to hash a message  $M$  onto the full domain  $Z_N^*$  of the RSA function before decrypting. The signature of  $M$  is  $f^{-1}(h(M))$ , where  $h$  is constructed to spread its argument uniformly into  $Z_N^*$ . According to the present invention, such a known technique is strengthened by making the hashing probabilistic. In order to sign message  $M$ , the signer first picks a random seed  $r$  of length  $k_0$ , where  $k_0 < k$  is a parameter of the scheme (recall  $k = |N|$ ). Then, using some hashing, in a specific way, the signer produces from  $M$  and  $r$  an image point  $y = \text{Hash}_{\text{PSS}}(r, M) \in$

$Z_N^*$ . As usual, the signature is then  $x = f^{-1}(y) = y^d \bmod N$ . Verification is more difficult, because one cannot simply recompute a probabilistic hash of  $M$  and expect to get the same value. Still, verification takes only one RSA encryption and  
5 some hashing, as will be seen below.

The inventive scheme is as efficient as known signing schemes based on RSA. But, it can be shown that the security of the inventive scheme is tightly related to that of RSA. Thus, for example, if the RSA inversion probability was originally  $2^{-61}$   
10 (using a certain amount of computational resources), then the probability of forgery for the signature scheme is almost equally low (assuming the same computational resources).

According to the invention, signing with "message recovery" is also provided. This technique reduces the bandwidth required for sending a signed message. In this technique, rather than transmit the message  $M$  and its signature  $x$ , a single enhanced signature  $\tau$ , of length less than  $|M| + |x|$ , is transmitted. The  
15 verifier is able to recover  $M$  from  $\tau$  and simultaneously check the authenticity. With security parameter  $k = 1024$ , the inventive  
20 scheme enables one to authenticate a message of up to, say,  $n = 767$  bits by transmitting only a total of  $k$  bits. The signing with message recovery scheme accomplishes this by appropriately folding the message into the signature in such a way that the



verifier can recover it. The computational efficiency and security are the same as for the first-described scheme.

In a further embodiment, the inventive technique is applied to the known Rabin function, with security tightly related to the  
5 hardness of factoring.

The foregoing has outlined some of the more pertinent objects and features of the present invention. These objects should be construed to be merely illustrative of some of the more prominent features and applications of the invention. Many other  
10 beneficial results can be attained by applying the disclosed invention in a different manner or modifying the invention as will be described. Accordingly, other objects and a fuller understanding of the invention may be had by referring to the following Detailed Description of the Preferred Embodiment.  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200  
205  
210  
215  
220  
225  
230  
235  
240  
245  
250  
255  
260  
265  
270  
275  
280  
285  
290  
295  
300  
305  
310  
315  
320  
325  
330  
335  
340  
345  
350  
355  
360  
365  
370  
375  
380  
385  
390  
395  
400  
405  
410  
415  
420  
425  
430  
435  
440  
445  
450  
455  
460  
465  
470  
475  
480  
485  
490  
495  
500  
505  
510  
515  
520  
525  
530  
535  
540  
545  
550  
555  
560  
565  
570  
575  
580  
585  
590  
595  
600  
605  
610  
615  
620  
625  
630  
635  
640  
645  
650  
655  
660  
665  
670  
675  
680  
685  
690  
695  
700  
705  
710  
715  
720  
725  
730  
735  
740  
745  
750  
755  
760  
765  
770  
775  
780  
785  
790  
795  
800  
805  
810  
815  
820  
825  
830  
835  
840  
845  
850  
855  
860  
865  
870  
875  
880  
885  
890  
895  
900  
905  
910  
915  
920  
925  
930  
935  
940  
945  
950  
955  
960  
965  
970  
975  
980  
985  
990  
995

## BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objects and advantages thereof, are best understood by reference to the following Detailed Description of an illustrative embodiment when read in conjunction with the accompanying Drawings, wherein:

**Figure 1** is a high level description of a preferred method of signing a message according to the present invention;

**Figure 2** is a diagram illustrating a preferred technique for generating an image point in a first signature scheme PSS of the present invention;

**Figure 3** is a diagram illustrating a preferred technique for generating an image point in a second signature scheme PSS-R of the present invention; and

**Figure 4** is a diagram illustrating a preferred technique for implementing the present invention with a Rabin signature scheme.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

By way of brief background, it is known that RSA is a family of *trapdoor permutations*. It is specified by the RSA generator, RSA, which, on input  $k$ , picks a pair of random distinct  $(k/2)$ -bit primes and multiplies them to produce a modulus  $N$ . It also picks, at random, an encryption exponent  $e \in Z_{\phi(N)}^*$  and computes the corresponding decryption exponent  $d$  so that  $ed \equiv 1 \pmod{\phi(N)}$ . The RSA generator returns  $N, e, d$ , these values specifying  $f: Z_N^* \rightarrow Z_N^*$  and  $f^{-1}: Z_N^* \rightarrow Z_N^*$ , which are defined by  $f(x) = x^e \pmod{N}$  and  $f^{-1}(y) = y^d \pmod{N}$ . Both functions are permutations, and, as the notation indicates, inverses of each other. The function  $f$  is the RSA encryption primitive and the function  $f^{-1}$  is the RSA decryption primitive.

The trapdoor permutation generator RSA-3 is identical to RSA except that the encryption exponent  $e$  is fixed to be 3. More generally, RSA- $e$  provides an encryption exponent of the specified constant. Other variants of RSA use a somewhat different distribution on the modulus  $N$ . The present invention, although stated for RSA, also hold for these other variants.

An inverting algorithm for RSA,  $I$ , gets input  $N, e, y$  and tries to find  $f^{-1}(y)$ . Its success probability is the probability that it outputs  $f^{-1}(y)$  when  $N, e, d$  are obtained by running  $RSA(k)$  and  $y$  is set to  $f(x)$  for an  $x$  chosen at random from  $Z_N^*$ .

The standard asymptotic definition of security requires that the success probability of any PPT (probabilistic, polynomial time) algorithm be a negligible function of  $k$ . The present invention goes further and, in particular, is interested in exactly how much time an inverting algorithm uses and what success probability it achieves in this time. Formally, an inverting algorithm is said to be a  $t$ -inverter, where  $t: IN \rightarrow IN$ , if its running time plus the size of its description is bounded by  $t(k)$ , in some fixed standard model of computation. The function  $I$   $(t, \epsilon)$ -breaks RSA, where  $\epsilon: IN \rightarrow [0, 1]$ , if  $I$  is a  $t$ -inverter and for each  $k$  the success probability of  $I$  is at least  $\epsilon(k)$ . The generator RSA is  $(t, \epsilon)$ -secure if there is no inverter which  $(t, \epsilon)$ -breaks RSA.

By way of further background, a digital signature scheme  $\Pi = (Gen, Sign, Verify)$  is specified by a key generation algorithm,  $Gen$ , a signing algorithm,  $Sign$ , and a verifying algorithm,  $Verify$ . The first two are probabilistic, and all three should run in expected polynomial time. Given  $k$ , the key generation algorithm outputs a pair of matching public and secret keys,  $(pk, sk)$ . The signing algorithm takes the message  $M$  to be signed and the secret key  $sk$ , and it returns a signature  $x = Sign_{sk}(M)$ . The algorithm may entail probabilistic choices. The verifying algorithm takes a message  $M$ , a candidate signature  $x'$ , and the

public key  $pk$ , and it returns a bit  $Verify_{pk}(M, x')$  with "1" signifying "accept" and "0" signifying "reject." If  $x$  was produced via  $x \leftarrow Sign_{sk}(M)$ , then  $Verify_{pk}(M, x) = 1$ .

One or more strong hash functions will usually be available to the algorithms *Sign* and *Verify*, with their domain and range depending on the scheme. According to the present invention, these functions are modeled as ideal, meaning that if hash function  $h$  is invoked on some input, the output is a uniformly distributed point of the range. If invoked twice on the same input, the same result is returned both times. In security proofs,  $h$  is modeled as a public random oracle (a "hash oracle") to be accessed via oracle queries, i.e., an algorithm can write a string  $z$  and get back  $h(z)$  in time  $|z|$ .

Definitions for the security of signatures in the asymptotic setting are known in the art. The following describes an "exact version" of these definitions. In particular, a forger takes as input a public key  $pk$ , where  $(pk, sk) \leftarrow Gen(k)$ , and tries to forge signatures with respect to  $pk$ . The forger is allowed a chosen message attack in which it can request, and obtain, signatures of messages of its choice. This is modeled by allowing the forger access to the signing algorithm. The forger is deemed successful if it outputs a valid forgery, namely, a message/signature pair  $(M, x)$  such that  $Verify_{pk}(M, x) = 1$  but  $M$  was not a message of which a signature was earlier requested.

The forger is said to be a  $(t, q_{\text{sig}}, q_{\text{hash}})$ -forger if its running time plus description size is bounded by  $t(k)$ ; it makes at most  $q_{\text{sig}}(k)$  queries of its signing oracle; and it makes a total of at most  $q_{\text{hash}}(k)$  queries of its various hash oracles. Such a forger  $F$  is said to  $(t, q_{\text{sig}}, q_{\text{hash}}, \epsilon)$ -break the signature scheme if, for every  $k$ , the probability that  $F$  outputs a valid forgery is at least  $\epsilon(k)$ . The signature scheme  $(\text{Gen}, \text{Sign}, \text{Verify})$  is  $(t, q_{\text{sig}}, q_{\text{hash}}, \epsilon)$ -secure if there is no forger who  $(t, q_{\text{sig}}, q_{\text{hash}}, \epsilon)$ -breaks the scheme.

Referring now to **Figure 1**, there is illustrated a high level description of a preferred method of signing a message according to the present invention. If  $M$  is the message desired to be signed, then the scheme begins by selecting a random, pseudorandom or otherwise time-varying seed value  $r$ . Such a value is sometimes referred to as a "nonce." A hash function then takes two arguments: the message  $M$  and the nonce  $r$ . The hash function then produces a keyed hash  $w = h(r, M)$ . As also seen in the figure, the message is split into two pieces, a first portion  $M_1$  and a second portion  $M_2$ , wherein the message  $M$  is easily recoverable from knowledge of the first and second portions. It is possible that the first portion or the second portion may take on a null value (or is "void"), such that the remaining portion is then the entire message.

Then, the signing routine encodes into an image point  $y$  (1) the hash value  $h(r,M)$ , (2) seed value  $r$ , and (3) the second portion  $M_2$  of the message  $M$ . These values are encoded in such a way so that  $r$ ,  $M_2$  and  $h(r,M)$  each are recoverable given an image point  $y$ . Then, the image point  $y$  is subjected to the decryption primitive  $f^{-1}$  to generate the signature  $x$  of the message  $M$ . The remaining portion  $M_1$  of the message (namely, that portion that was not encoded into the image point) is then concatenated or otherwise combined with the signature to form the enhanced signature  $x$ .

Thus, according to the invention, the message  $M$  to be signed is decomposed or otherwise split into first and second portions. One of the portions, in effect, is "folded" into the trapdoor permutation, while the other portion (i.e., the portion that does not fit) gets transmitted with the signature  $x$  to facilitate the authentication. Thus, the signature scheme is a combined "signature with appendix" and "signature permitting message recovery" scheme within the meaning of ISO/IEC 9796 and ISO/IEC 14888-1.

A keyed hashing function of  $M$  is used with a seed value  $r$ , wherein the seed value is communicated in some way in the image point. Preferably, a random seed value  $r$  is used for each message. The image point is unpredictable due to the seed value.

Referring now to **Figure 2**, a first detailed embodiment of the inventive probabilistic signature scheme is now illustrated and described. The scheme  $PSS[k_0, k_1]$  is described by a key generation algorithm  $GenPSS$ , a signing algorithm  $SignPSS$  and a verifying algorithm  $VerifyPSS$ . The latter two algorithms are parameterized by  $k_0$  and  $k_1$ , which are numbers between 1 and  $k$  satisfying the relationship  $k_0 + k_1 \leq k - 1$ . Thus, for example,  $k = 1024$ , and  $k_0 = k_1 = 128$ . Such values, of course, are merely representative. The key generation algorithm  $GenPSS$  runs  $RSA(k)$  to obtain  $(N, e, d)$ , and outputs  $(pk, sk)$ , where the public key is  $pk = (N, e)$  and the secret key is  $sk = (N, d)$ .

As illustrated in **Figure 2**, the signing and verifying algorithms preferably make use of two hash functions. The first,  $h$ , sometimes referred to as a compressor, maps as  $h: \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^{k_1}$  and the second,  $g$ , sometimes referred as the generator, maps as  $g: \{0, 1\}^{k_1} \rightarrow \{0, 1\}^{k-k_1-1}$ . These functions are preferably implemented from conventional cryptographic hash functions, such as MD5 or SHA-1, as will be described below. For purposes of the scheme,  $g_1$  is a function that, on input  $w \in \{0, 1\}^{k_1}$ , returns the first  $k_0$  bits of  $g(w)$ , and  $g_2$  is a function that, on input  $w \in \{0, 1\}^{k_1}$ , returns the remaining  $k - k_0 - k_1 - 1$  bits of  $g(w)$ . The signature  $x$  of a message  $M$  (sometimes referred to herein as a *data string*) is then:



*SignPSS(M)*

$r \leftarrow \{0,1\}^{k_0}$  ;  $w \leftarrow h(r,M)$  ;  $r^* \leftarrow g_1(w) \oplus r$

$y \leftarrow 0 \parallel w \parallel r^* \parallel g_2(w)$

return  $y^d \bmod N$

- 5 Thus, given the message  $M$  and signature  $x$ , verification proceeds as follows:

*VerifyPSS*

$y \leftarrow x^e \bmod N$

Let  $y$  be decomposed as:  $b \parallel w \parallel r^* \parallel \gamma$  ,

(where  $b$  is the first bit of  $y$ ,  $w$  the next  $k_1$  bits,  $r^*$  the next  $k_0$  bits, and  $\gamma$  the remaining bits)

$r \leftarrow r^* \oplus g_1(w)$

if  $(h(r,M) = w$  and  $g_2(w) = \gamma$  and  $b = 0)$ , then return **1**;  
otherwise, return **0**.

- 10 Above, the step  $r \leftarrow \{0,1\}^{k_0}$  indicates that the signer picks (preferably at random) a seed  $r$  of  $k_0$  bits. The signer then hashes this seed with the message  $M$ , for example, by concatenating these strings and applying some cryptographic hash function (such as SHA-1). The result is a  $k_1$ -bit string  $w$ .
- 20 Then, the generator  $g$  is applied to  $w$  to yield a  $k_0$ -bit string  $g_1(w)$  and a  $k-k_0-k_1-1$  bit string  $g_2(w)$ . The first bit string is then used to "mask" the  $k_0$ -bit seed  $r$ , resulting in the masked

seed  $r^*$ . Then,  $w|r^*$  is prepended with a 0 bit and appended with  $g_2(w)$  to create the image point  $y$ , which is decrypted under the RSA function to yield the signature  $x$ . The 0-bit is to substantially guarantee that  $y$  is in  $Z_N^*$ . Preferably, a new seed  
 5 is chosen for each message. In particular, a given message has many possible signatures, depending on the value of  $r$  chosen by the signer.

Given  $(M, x)$ , the verifier first computes  $y = x^e \bmod N$  and recovers  $r^*$ ,  $w$  and  $r$ . These values are then used to check that  $y$   
 10 was correctly constructed, and the verifier only accepts the message if all the checks succeed. Thus, signing takes one application of  $h$ , one application of  $g$ , and one RSA decryption, while verification takes one application of  $h$ , one application of  $g$ , and one RSA encryption. Thus, the scheme is quite efficient.

Another embodiment of the invention is illustrated in **Figure 3**, and is now described. By way of brief background, in a standard signature scheme, the signer transmits the message  $M$  in the clear, attaching to it the signature  $x$ . In a scheme that provides "message recovery," only an "enhanced signature"  $x$  is  
 20 transmitted. The goal is to save on the bandwidth for sending a signed message. In particular, it is desired that the length of this enhanced signature to be smaller than  $|M| + k$ . When  $M$  is short enough, it is desired that the length of  $x$  be  $k$ , the

signature length. The verifier then recovers the message  $M$  from the enhanced signature and checks authenticity at the same time.

Signing with message recovery is accomplished according to the present invention by "folding" some or all of the message into the signature in such a way that it is "recoverable" by the verifier. When the length  $n$  of  $M$  is small, the entire message can be folded into the signature, so that only a  $k$ -bit quantity is transmitted. In the preferred scheme defined below, if the security parameter is  $k = 1024$ ,  $k_0 = 128$ , and  $k_1 = 128$ , one can fold up to **767** message bits into the signature. This value, of course, is merely representative.

In a signature scheme permitting message recovery, the definition of the key generation and signing algorithms are as described previously, but the verification algorithm is replaced by a "recovery algorithm" which takes the public key  $pk$  and the enhanced signature  $x$  and returns  $Recover_{pk}(x) \in \{0,1\}^* \cup \{REJECT\}$ . The distinguished point REJECT is used to indicate that the recipient rejected the signature; a return value of  $M \in \{0,1\}^*$  indicates that the verifier accepts the message  $M$  as authentic. The formulation of security is the same as that above except for what it means for the forger to be successful. In particular, it should provide an  $x$  such that  $M = Recover_{pk}(x) \in \{0,1\}^*$ , where  $M$  was not a previous signing query. If  $x$  was produced via  $x \leftarrow Sign_{sk}(M)$ , then  $Recover_{pk}(x) = M$ .

A simple variant of the PSS scheme achieves message recovery. The scheme PSS-R  $[k_0, k_1]$  includes the same key generation algorithm GenPSS as previously described. As with PSS, the signing and verification algorithms depend on hash function  $h: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^{k_1}$  and  $g: \{0,1\}^* \rightarrow \{0,1\}^{k-k_1-1}$ , and the same  $g_1$  and  $g_2$  notation is used as before. For simplicity of explanation, it is assumed that the messages to be signed have length exactly  $n = k - k_0 - k_1 - 1$ . Thus, possible choices of parameters are  $k = 1024$ , with  $k_0 = k_1 = 128$  and  $n = 767$ . As previously noted, these values are merely representative. In this embodiment, an enhanced signature of only  $k$  bits is generated from which the verifier can recover the  $n$ -bit message and simultaneously check authenticity. Signature generation and verification proceed as follows:

*SignPSS-R(M)*

$r \leftarrow \{0,1\}^{k_0} ; w \leftarrow h(r, M) ; r^* \leftarrow g_1(w) \oplus r$

$M^* \leftarrow g_2(w) \oplus M$

$y \leftarrow 0 \parallel w \parallel r^* \parallel M^*$

return  $y^d \bmod N$

*RecoverPSS-R(x)*

$y \leftarrow x^e \bmod N$

Let  $y$  be decomposed into:  $b \parallel w \parallel r^* \parallel M^*$ ,

(where  $b$  is the first bit of  $y$ ,  $w$  the next  $k_1$  bits,  $n^*$  the next  $k_0$  bits, and  $M$  the remaining  $r$  bits)

$$r \leftarrow r^* \oplus g_1(w)$$

$$M \leftarrow M^* \oplus g_2(w)$$

5       if ( $h(r, M) = w$  and  $b = 0$ ), then return  $M$ ; otherwise, return REJECT.

Thus, *SignPSS-R* differs with respect to the signature algorithm of the first embodiment (*SignPSS*) in that the last part of the image  $y$  is not  $g_2(w)$ . Instead,  $g_2(w)$  is used to "mask" the message, and the masked message  $M^*$  is the last part of the image point  $y$ . This scheme is then easily adapted to handle messages of arbitrary length. A fully-specified scheme preferably would use about  $\min\{k, n + k_0 + k_1 + 16\}$  bits for the enhanced signature of the  $n$ -bit message  $M$ .

16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
The present invention extends to Rabin signatures, yielding a signature scheme and a signing with recovery scheme whose security can be tightly related to the hardness of factoring. In the basic Rabin signature scheme described in *Digitalized Signatures and Public Key Functions as Intractable as*  
20 *Factorization*, MIT/LCS/TR-221, January 1979, the signer chooses a number  $N = pq$  that is the product of two large primes, and the signer further chooses a number  $0 \leq b < N$  (for example, the signer may choose  $b = 0$ ). The signer's public key is  $(N, b)$  and

the corresponding secret key is  $(p, q, b)$ . To sign a message  $M$ , the signer chooses a random seed  $r$  and computes a hash  $w = h(r, M)$ . The point  $w$  is regarded as a value in  $Z_N^*$ , the multiplicative group of integers modulo  $N$ . The signer then checks if the equation  $x(x+b) = w$  has a solution in the multiplicative group of integers modulo  $N$ . This can be done using the signer's private key in a manner well-known in the art and described by Rabin. If the above equation has no solution, then the signer chooses a new random seed  $r$  and repeats the process described above. If the equation does have a solution the signer chooses one such solution, call it  $x$ , and the signature of  $M$  is taken to be the ordered pair  $(x, r)$ .

An analogous scheme is implemented in the present invention and has equally efficient computation time but shorter signatures (because there is no need to separately transmit the seed  $r$  as part of the signature).

The probabilistic Rabin scheme is defined by  $PRab[k_0, k_1] = (GenPRab, SignPRab, VerifyPRab)$ , and depends on parameters  $k_0, k_1$ , where  $k_0 + k_1 \leq k$ . Algorithm  $GenPRab$ , on input  $k$ , picks a pair of random distinct  $(k/2)$ -bit primes  $p, q$  and multiplies them to produce the  $k$ -bit modulus  $N$ . It outputs  $(pk, sk)$ , where  $pk = N$  and  $sk = (N, p, q)$ .

The signing and verifying algorithms of  $PRab$  preferably use hash functions  $h, g$ , where  $h: \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^{k_1}$  and  $g$ :

$\{0,1\}^{k_1} \rightarrow \{0,1\}^{k-k_1}$ . In particular, let  $g_1$  be the function that on input  $w \in \{0,1\}^{k_1}$  returns the first  $k_0$  bits of  $g(w)$ , and let  $g_2$  be the function that on input  $w \in \{0,1\}^{k_1}$  returns the remaining  $k - k_0 - k_1$  bits of  $g(w)$ .

5       The signing procedure, *SignPRab*, is similar to the corresponding *SignPSS*, but this routine returns a random square root of the image  $y$ , as opposed to  $y^d \bmod N$ . The verification procedure checks if the square of the signature has the correct image. Thus verification is particularly fast. Here, in full, are preferred forms of *SignPRab* (illustrated in **Figure 4**) and *VerifyPRab*:

*SignPRab* ( $M$ )

    repeat

$r \leftarrow \{0,1\}^{k_0}; w \leftarrow h(r,M); r^* \leftarrow g_1(w) \oplus r$

$y \leftarrow w \parallel r^* \parallel g_2(w)$

    until  $y \in Z_N^*$  and  $y$  is a quadratic residue mod  $N$ .

    Let  $\{x_1, x_2, x_3, x_4\}$  be the four distinct square roots of  $y$  in  $Z_N^*$ .

    Choose  $x \leftarrow \{x_1, x_2, x_3, x_4\}$  at random.

20     return  $x$ .

*VerifyPRab* ( $M, x$ )

$y \leftarrow x^2 \bmod N$

Break up  $y$  as  $w \parallel r^* \parallel Y$ . (That is, let  $w$  be the first  $k_1$  bits of  $y$ ,  $r^*$  the next  $k_0$  bits, and  $Y$  the remaining bits.)

$r \leftarrow r^* \oplus g_1(w)$

if  $(h(r, M) = w \text{ and } g_2(w) = Y)$  then return **1**; else return **0**.

5 As with PSS, one can add message recovery to the Rabin scheme in the same way, resulting in a probabilistic Rabin signing-with-recovery scheme, PRab-R. Its security is the same as that of the Rabin-based scheme.

Thus, in the Rabin schemes, the seed value  $r$  is chosen repeatedly until the image string  $y$  is in the domain of the decryption primitive and the primitive is the probabilistic function that returns a random square root of the image point in the multiplicative group of integers modulo a product of two primes. Thus, a "Rabin decryption primitive" is a map which takes a square  $x^2$  in  $Z_N^*$  and returns a random square root of  $x^2$ . The seed itself is encoded into the image point so that it does not have to be transmitted separately from the decrypted image point.

Note that an RSA-based scheme could take on a similar form,  
20 where the seed value is chosen repeatedly until  $y$  is in the domain; in such case, the leading "0" bit of  $y$  (as described above) would not be necessary. Moreover, variants of Rabin encryption, such as those described in ISO/IEC **9796**, could be



applied (in the Rabin scheme above) to reduce the number of iterations.

The previously described PSS, PSS-R, Prab, and PRab-R schemes require a concrete hash function  $h$  with output length some given number  $k_1$ . Typically,  $H$  may be constructed from some cryptographic hash function  $H$  such as  $H = \text{MD5}$  or  $H = \text{SHA-1}$ . A simple technique is to define  $h(x)$  as the appropriate-length prefix of:

$$h(\text{const.}\langle 0 \rangle.x \mid h(\text{const.}\langle 1 \rangle.x \mid h(\text{const.}\langle 2 \rangle.x \mid \dots$$

The constant  $\text{const}$  should be unique to  $h$ . To make another hash function,  $g$ , one simply selects a different constant. Many similar such constructions are possible of course.

The present invention provides numerous advantages. It can be shown that the security of the PSS (or PSS-R) is based on the security of RSA, but with a relationship between the two that is much "tighter" than in the prior art. In particular, it has been shown that if the RSA generator is  $(t', \epsilon')$  secure, then for any  $q_{\text{sig}}, q_{\text{hash}}$ , the signature scheme  $\text{PSS}[k_0, k_1]$  (or PSS-R) is  $(t, q_{\text{sig}}, q_{\text{hash}}, \epsilon)$ -secure, where:

$$t(k) = t'(k) - [q_{\text{sig}}(k) + q_{\text{hash}}(k) + 1] \cdot k_0 \cdot \Theta(k^3), \text{ and}$$

$$\epsilon(k) = \epsilon'(k) + [3(q_{\text{sig}}(k) + q_{\text{hash}}(k))^2] \cdot (2^{-k_0} + 2^{-k_1}).$$

With respect to the Rabin probabilistic scheme, it has been shown that if factoring is  $(t', e')$ -hard, then, for any  $q_{sig}$ ,  $q_{hash}$ , the signature scheme PRab[ $k_0, k_1$ ] is  $(t, q_{sig}, q_{hash}, \epsilon)$ -secure, where:

$$t(k) = t'(k) - [q_{sig}(k) + q_{hash}(k) + 1] \cdot k_0 \cdot \Theta(k^2), \text{ and}$$

$$\epsilon(k) = 2\epsilon'(k) + [6(q_{sig}(k) + q_{hash}(k))^2] \cdot (2^{-k_0} + 2^{-k_1}).$$

One of the preferred implementation of the probabilistic signature scheme of the present invention is a computer program. One implementation is in conjunction with a development toolkit. Thus, each of the algorithms of the invention are preferably implemented as a set of instructions (program code or instruction means) in a code module resident in the random access memory (RAM) of a computer. The set of instructions (or some portion thereof) may be stored in another computer memory or downloaded via the Internet or other computer network.

The computer program comprising the probabilistic signature scheme is executed in a computer. The computer used in the present invention is any personal computer or workstation client or server platform that is Intel-, PowerPC®- or RISC®-based, and that includes an operating system such as IBM® OS/2®, Microsoft Windows 95, Microsoft Windows NT 4.0, Unix, AIX®, OS/400 or the like.

Although not required, the various processing routines that comprise the present invention may reside on the same host machine or on different machines interconnected over a network

(e.g., the Internet, an intranet, a wide area network (WAN) or local area network (LAN)). Thus, for example, the signature of the message may be performed on one machine, with the associated verification then performed on another machine. Thus, a computer  
5 running the present invention has appropriate networking hardware to establish a connection to another computer in a conventional manner.

10 In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

15 The particular message "content" is not a limitation of the present invention. Thus, the message may be generalized as any "data string" irrespective of the particular application for which the digital signature scheme is to be used.

20 Having thus described our invention, what we claim as new and desire to secure by Letters Patent is set forth in the following claims.